"AN APPARATUS FOR MIGRATION AND CONVERSION OF SOFTWARE CODE FROM ANY SOURCE PLATFORM TO ANY TARGET PLATFORM"

This invention relates to 'An Apparatus to create automation tools for migration and conversion of software code from any source platform to any target platform'. More particularly this invention relates to an apparatus for migration and conversion of software code using a knowledge engine to identify the source application systematically and logically and to convert the same logic and database to any target application using target - specific knowledge base.

The existing systems for migration and conversion of software have lots of drawbacks in various areas such as application, limitations on conversion weighting, limited compatibility, etc.
Such as,

- Currently, the existing system for migration or updation of any legacy application developed, using a known technology is achieved manually. Hence, the amount of time spent and the cost invested for the same are very high.
- Moreover, there is no specific tool available to migrate and/or upgrade any application. There may be an individual tool for each such conversion.
- For each and every source application if such a tool/service exists then that has to be purchased individually, which makes the migration very costly.
- For each migration as it is a new process in existing system conversion time will have no effect of previous conversion processes. Whereas in applicant's invention, more the iterations, the higher is the quality & speed of conversion that is achieved due to the applicant's invention being a Knowledge based system.

Objects of the Invention: -

It is clear from the above explanations that the current systems are very difficult to be reused or reapplied everywhere and there is no unique solution for migration and conversion. So with the motto of making an apparatus, which can migrate and convert any source application of any source platform to any target platform, the inventor has developed a system or technology named 'An APPARATUS FOR MIGRATION AND CONVERSION OF SOFTWARE CODE' by which, conversion and/or migration of any application becomes possible with best results.

The main object of the invention is to make the apparatus for conversion and/or migration of any source application with no limitation of platforms on which the source or target systems are working.

The further object of the invention is to make the apparatus, which uses a knowledge base concept, such that conversion process time decreases as more source applications are processed by the applicant's software tool.

The further object of the invention is to make an apparatus, which will take the possibility of future conversions and/or migrations into account by using a knowledgebase for better and speedy conversions in future.

The further object of the invention is to create effective tools for current and future system migrations, design and development.

The further object of the invention is to reduce the overall development effort by having the tools, which learn from the created knowledge database, automate the migration, design and development processes as much as possible.

The further object of the invention is to segregate and assemble process knowledge and attributes in a structured format to create a knowledge database for future use.

BRIEF DESCRIPTION OF THE DRAWING

FIG-1 shows the block diagram of the system

FIG-2 shows the Aided process for migration or enhancement of an applic

Description of the System: -

According to the invention, an 'Apparatus for Migration and Conversion o Software Code from Any Source Platform to Any Target Platform' migrates and/or convert any source application working on any platform into a format that any target platform comprises of.

An Inputting means for accepting the entire source code in ASCII format tc analyse the business logic of the source application, UI (User Interface)/GUI (Graphical Use Interface) details of the source and target application, validation schemes of source front end interface, definitions of the target back-end system, existing test scripts to facilitate the quality control phase of the generated code, the source code entry points to busines processes and target environment specification or definitions which include targe platform(s),languages to be used, target database, coding standards, target architecture and framework, third party components, details of existing applications which have to be plugged into the target application and sample code of the application working in targe environment (if available);

An Analysing means for analysing the source schemes provided by client to create target schemas, analysing the workflow diagrams that represent the source applicatior process, identifying the code segments in source application and analysing the targe requirements that will generate the target architecture and the technology associated with it;

A Setting up means for generating custom knowledge base where the existing KB i reviewed for particular migration and in case no such KB exists, a custom KB is created;

A Processing means for conversion of source code into format of targe specification wherein complete source code is passed through a knowledge engine on the basis of multiple iterations and during this time the knowledge engine remains coupled to the knowledge base for conversion of source code into format of target specification. After each iteration the knowledge base is updated which results in speedy and better conversion o

source code as the Custom KB gets more structured with respect to source platform, source application, target platform and target application; and

A Documenting means for generation of reports related to process stage and end of conversion process. This consists of the code that is not converted automatically, the reasons for the non-conversion and suggestions on how the unconverted code can be converted manually at applicants Resource Centre.

The system is divided into two main parts: -

1. Knowledge Base, which is herein after referred as 'KB'
2. Knowledge Engine, which is herein after referred as 'KE'

Knowledge Base: - The custom KB as shown block M in fig. 1 consists of a relational database that consists of source and target code patterns and attributes. The custom KB (block M resembles and consolidates the source and target language syntax, control structure and programming style. The structure of the custom KB (block M) is decided on the basis of forma of source and target application. The custom KB (block M) is updated after every iteration by 'KE', which increases the rate of automatic conversion.

The KB can be accessed by standard ODBC (Object Data Base Connectivity connection or through disk I/O (Input/Output) operations. The KB may also involve a interface layer that will allow the KE to connect to it remotely.

Knowledge Engine: - The KE shown as block N in fig. 1 is a processing unit which allows the input code to be intercepted, interpreted and converted into target output code. I automates code migration and enhancements, allowing for many different design and development options.

To extract the business logic from the source application it does the following functions:

- Intercept the input code based on defined syntax in the custom KB (block M).
- Interpret the source code using fuzzy logic routines.
- Parse input code, segregate code blocks and convert them into intermediate or targe specific format using custom KB (block M)

- Marks the unconverted or ignored blocks and sentences of the source code and stores them in the custom KB (block M) for future use.

In case of migration of a data driven system the KE will work as follows:-

- Interpret database schemes and data file formats.
- Generate KB (block M) segment – thereby creating data dictionary definitions.
- Interpret source code, parse it and convert it into target specific code with the help of data dictionary definitions generated in the previous step.
- Marks the unconverted or ignored blocks and sentences of the source code and stores them in the custom KB (block M) for future use.
- Repeat the above process for arriving at the final code thereby involving intermediate processes.

IN BLOCK S

Inputs a, b, c, d and e represent the input code components. These components are the input files that are divided based on Program Logic Code, User Interface, Data Dictionary Validation, Associated Program Logic files or Data Definition files. The user performs this task.

IN BLOCK 'T' F IS Code validation stage. Here the KE determines if all the required input files are available for processing

IN BLOCK U

N IS This is the Knowledge Engine (KE) component of the S2T Technology. This component is the primary processing module that processes the input code

M is the Knowledge Base (KB) component of the S2T Technology that stores patterns and discovered data.

L symbol represents the "iterative process" by which input code is iteratively processed by KE.

O component is logical extension of the KE and is used to package the output code generated by KE.

IN BLOCK V

Q block represents the target code generated by the tool as well as the Summary Report documenting the entire conversion process.

P block represents the ignored or un-converted source code. The ignored code is fed back into the KB after verification to determine new patterns, which the user will insert into the KB through the KB configuration process

Working of the System: -

The working of the invention is briefly explained with the help of drawings as under:

When any source application is required to be converted and/or migrated to a target specific format then procedure will be as described under: -

The first stage named 'Input Stage' is shown in fig 1 as block S for migration or conversion process. This stage includes receiving the source application to analyse it systematically and logically. It also includes defining the architecture of the target application.

After completing this task , either an existing library is refined or a new library custom KB (block M) is created to fulfil the requirements of the source and target application as closely as possible.

These following inputs are received at the Input Stage:

1. The entire source code of sample part in ASCII (American Standard Code for Information Interchange), shown as 'a' is received so it can be analysed to understand the business logic of the source application, upon which custom KB (block M) is defined i.e. created or updated.

2. The UI (User Interface)/GUI (Graphical User Interface) details of the source application are received and the UI details of the target application are defined. This is shown as 'b'. For an effective conversion, validation schemes of source front-end interface are also received into this stage.

3. The definitions of the target system database shown as 'c' in the figure are obtained. For a relational target database the scheme formats may be in SQL (Structure Query Language) or, for mainframe computers, in fragments of the flat file database with copybook structure in ASCII format.

4. If there already exists test scripts shown as 'd' in the figure, they are received and used to facilitate the quality control phase of the generated target code.

5. It is necessary to determine the source code entry points to business processes shown as 'e'. In some cases details provided by the UI may be sufficient to decide the entry points. If not, the user is asked to specify the entry points.

6. The target environment specification or definitions are defined, including
    - Target platform(s)
    - Languages to be used
    - Target database
    - Coding standards
    - Target architecture and framework
    - Third party components
    - Existing applications which have to be plugged with target application
    - Sample code for the application working in target environment (if available)

The second stage is the Analysis Stage, which is shown as block T in the figure in which the source code is analysed for conversion. In this stage, first the database analysis is done in which the source schemas provided by client is analysed to create target schemas. Then the business analysis is done in which workflow diagrams that represent the source application processes are obtained. This analysis is used in generating the custom KB (block M). The last process of this stage is target analysis that generates the target architecture and the technology associated with it, which are added to the target application segment of the custom KB (block M).

The third stage is 'Setup Stage' shown as block U in which generation of custom KB (block M) takes place. First the existing KB (block M), is reviewed for particular migration. In case of no such KB (block M) exists, a custom KB (block M) is created as follows:

The selected parts of sample code 'a', UI 'b' and database (back-end schemes) 'c' are introduced to KE (block N). The formulation done by KE (block N) on 'a', 'b' and 'c' form the base for the target requirements in the custom KB (block M). Through an iterative process shown as 'L', in which the selected source code fragment is passed multiple times through to KE (block N), knowledge patterns are collected. This process continues until the KE (block N) gets the signal for saturation from the KB (block M). In case of test scripts provided by the client, they are passed through KE (block N) also; otherwise the test scripts created by the KE (block N) itself.

In the next stage called 'Process Stage' shown as block U, *the* complete source code is passed through the KE (block N) on the basis of iteration. During this time the KE (block N) is coupled to the Custom KB (block M) for conversion of source code in the format of target specifications. After each iteration, Custom KB (block M) is updated which leads to speedy and better conversion of source code as the Custom KB (block M) has now more structured information of source platform and source application with respect to target platform and target application.

The last stage is the documentation stage shown as block V, in which the reports generated during process stage are reviewed. After the conversion process a summary report shown as P is generated, which consists of the code that is not converted (block O) automatically. This unconverted code is then converted manually at applicants Resource Centre. However, by this stage applicant has achieved 70 to 90% automatic conversion. In this stage two other processes are also performed namely:

- Target Database Verification, which includes verifying the converted database and the data dictionary with respect to the source scheme.
- Target Application Verification, which includes verifying the converted application and its program links with respect to the source code process flow during the Analysis Stage (block S)

The description of the system given above is given only to understand the invention rather than to limit its scope.